8K

УДК 004.89

М.А. Князева, И.И. Москалев
Институт автоматики и процессов управления ДВО РАН, г. Владивосток, Россия
mak@imcs.dvgu.ru, mosk85@mail.ru

Интерпретатор методов потокового анализа программ в специализированном банке знаний о преобразованиях программ*

Интерпретатор методов потокового анализа является основной частью подсистемы потокового анализа в специализированном банке знаний о преобразованиях программ (СБкЗ_ПП). Данная подсистема осуществляет интерпретацию методов потокового анализа и занимается расширением модели структурных программ, терминами потокового анализа. Расширение модели программы терминами потокового анализа необходимо для дальнейшего проведения оптимизирующих преобразований на ней. Статья содержит описание методов потокового анализа программ в терминах онтологии знаний о потоковом анализе программ, метода интерпретации методов потокового анализа программ и описание интерпретатора методов потокового анализа программ.

Введение

Эквивалентность преобразований программ в системах преобразований программ отслеживается с помощью какой-либо модели информационных зависимостей.

В работе В.В. Воеводина [1] предложен метод графа алгоритма (или решетчатого графа), который был реализован в системе V-Ray [2], предназначенной для выявления возможностей параллелизма в последовательных ФОРТРАН – программах. Основным понятием V-Ray начальной вершине, используется в качестве аргумента операции, соответствующей технологии, является граф алгоритма – ациклический ориентированный граф, вершины которого соответствуют отдельным срабатываниям операторов программы. Две вершины графа соединяются направленной дугой, если результат операции, соответствующей конечной вершине. Таким образом, понятие графа алгоритма опирается на два основных понятия: отдельные срабатывания операторов программы и точное описание информационных зависимостей между ними. Система позволяет также построить граф зависимости по данным по всем четырем типам зависимостей или их комбинациям. Граф может быть построен с использованием графа алгоритма, построенного с помощью V-Ray технологии или без его использования, так как не всегда возможно построение графа алгоритма.

В работе [3] рассматривается проект системы ПРОГРЕСС. Совокупности преобразований, поддерживаемые системой ПРОГРЕСС, должны базироваться на богатой библиотеке преобразований, содержащей базовые преобразования для разных промежуточных представлений программ, и развитых средствах конструирования систем преобразований. Библиотека преобразований включает наряду с универсальными (машинно-независимыми) и машинно-зависимые преобразования, как правило, ориентированные на те или

---

*Работа выполнена в рамках программы №14 фундаментальных исследований Президиума РАН, проект 06-I-П14-052.

иные классы целевых архитектур. Средства конструирования систем преобразований содержат встроенные проверки контекстных условий и стратегии применения наборов преобразований, а также поддерживают возможность их расширения и задания новых стратегий применения преобразований.

В открытой распараллеливающей системе ОРС [4] библиотека преобразований выделяется в отдельную самостоятельную часть. Такая самостоятельность позволяет библиотеке уделять больше внимания, преобразования классифицировать и группировать, сформировать специальную архитектуру библиотеки, систему тестов для преобразований.

Подход, реализованный в системе БЕТА, заключается в том, что программа в процессе анализа пополняется так называемыми тенями [5]. Тени содержат иерархическое представление управляющего графа программы подграфами специального вида. Такими подграфами являются зоны или сильно связанные подграфы гамаки или такие подграфы, которые имеют единственную входную вершину, принадлежащую гамаку, и единственную выходную вершину, ему не принадлежащую, и линейные компоненты зон или такие минимальные гамаки в зоне, которые исполняются всегда при исполнении зоны.

Во всех рассмотренных системах реализован тот или иной класс моделей информационных зависимостей. В работе предложен подход, в котором возможно представлять в виде онтологии методов потокового анализа программ различные информационные модели, записывать их на специально разработанном языке в базу знаний методов потокового анализа программ. В отличие от существующих подходов в системах, знания о методах потокового анализа в предложенной концепции представляются в явном виде в декларативной форме.

Методы потокового анализа программ в терминах онтологии знаний о потоковом анализе программ

Как правило, в процессе трансляции программа представляется в виде некоторой модели или внутреннего представления. В работе в качестве модели внутреннего представления программы используется модель структурных программ (МСП) [6]. В работе [7] предложен специальный язык для записи методов потокового анализа программ.

Язык описания методов потокового анализа программ (язык МПА) разработан с целью формализации процесса расширения МСП. При разработке языка были проанализированы основные формы записи методов потокового анализа, описанные в литературе [2], [3], [5] .

Язык позволяет описывать конструкции, которые являются результатом потокового анализа программы: атрибуты вершин графа; отношения и функции над вершинами графа; вершины и дуги графа.

Язык МПА содержит переменные, которые в качестве значений могут принимать ссылки на различные элементы модели программы; основные конструкции алгоритмических языков программирования (такие, как цикл, выбор, присваивание); операции с множествами, поскольку в процессе накопления информации о программе часто оперируют с множествами переменных и идентификаторов; операции обходов деревьев.

База знаний о методах потокового анализа программ формируется экспертом предметной области. В нее могут входить как методы, взятые из различных источников (описанные в литературе, в сети Интернет и т.д.), так и методы, разработанные непосредственно экспертом.

Поскольку для потокового анализа модель программы уже является фиксированной, то один из вариантов представления информации о потоковом анализе – это добавление к этой модели атрибутов, смысл которых бы и заключался в отображении информации о потоковом анализе. Множество таких атрибутов называется расширением МСП терминами потокового анализа. Это расширение описано в [8].

Существуют всевозможные каталоги, в которых описаны методы преобразования программ. В рамках данного подхода знания о преобразованиях хранятся в унифицированной форме в каталоге на специальном языке представления знаний. В описаниях таких преобразований присутствует информация, относящаяся к методам потокового анализа программ. Для представления методов потокового анализа в банке знаний был разработан язык представления методов потокового анализа программ. Далее представлено неформальное описание одного из методов потокового анализа программ.

Метод вычисления аргументного, результатного и сильнорезультатного информационных множеств для фрагментов модели структурной программы класса Выражение

Для всех фрагментов модели структурной программы класса *Выражение*, у которых левая часть является константой или переменной:

1. *Аргументное*, *результатное и сильнорезультатное* множества, равны соответствующим множествам правой части.

2. Если левая часть является переменной, то она добавляется к *аргументному* множеству (рис. 1, рис. 2).

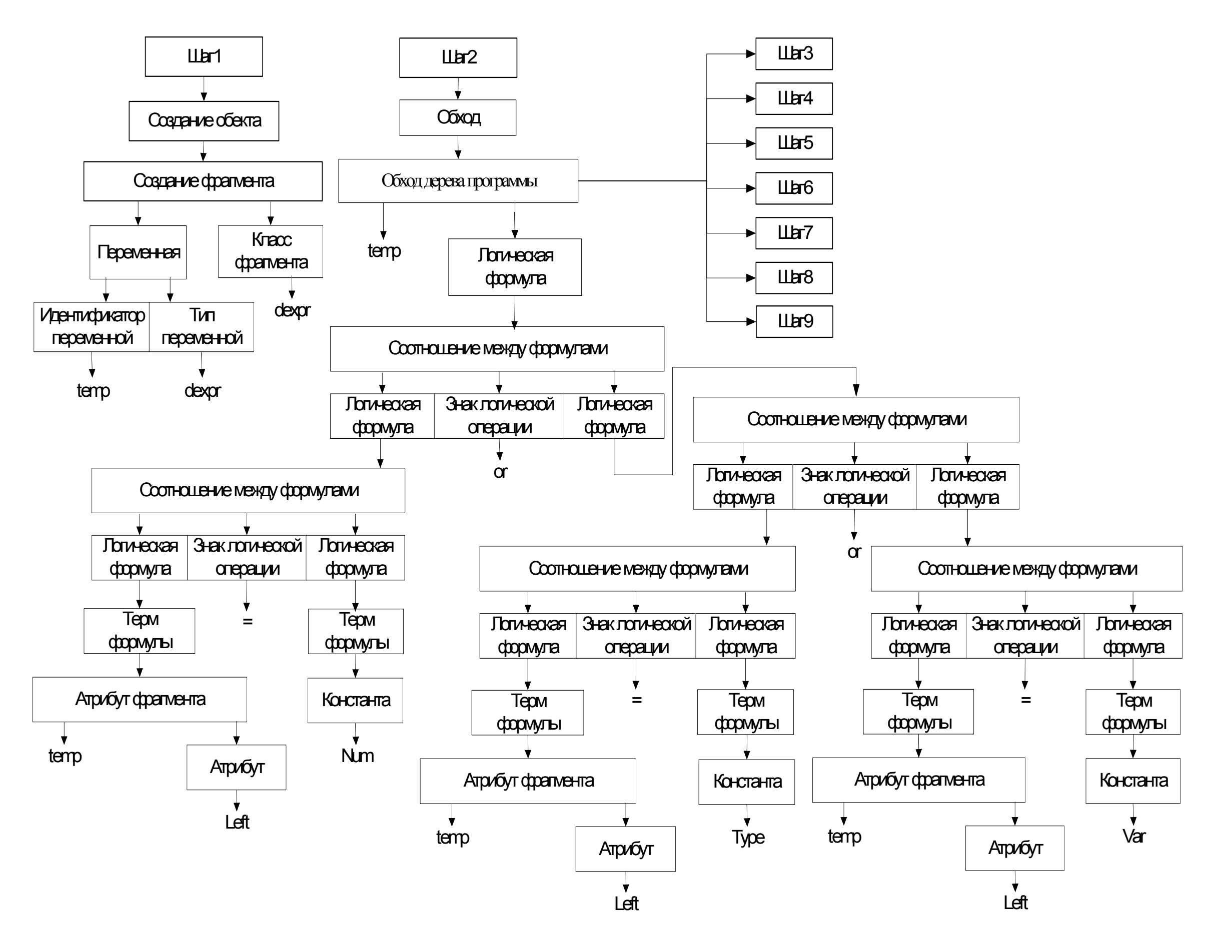


Рисунок 1 – Формальное представление метода потокового анализа

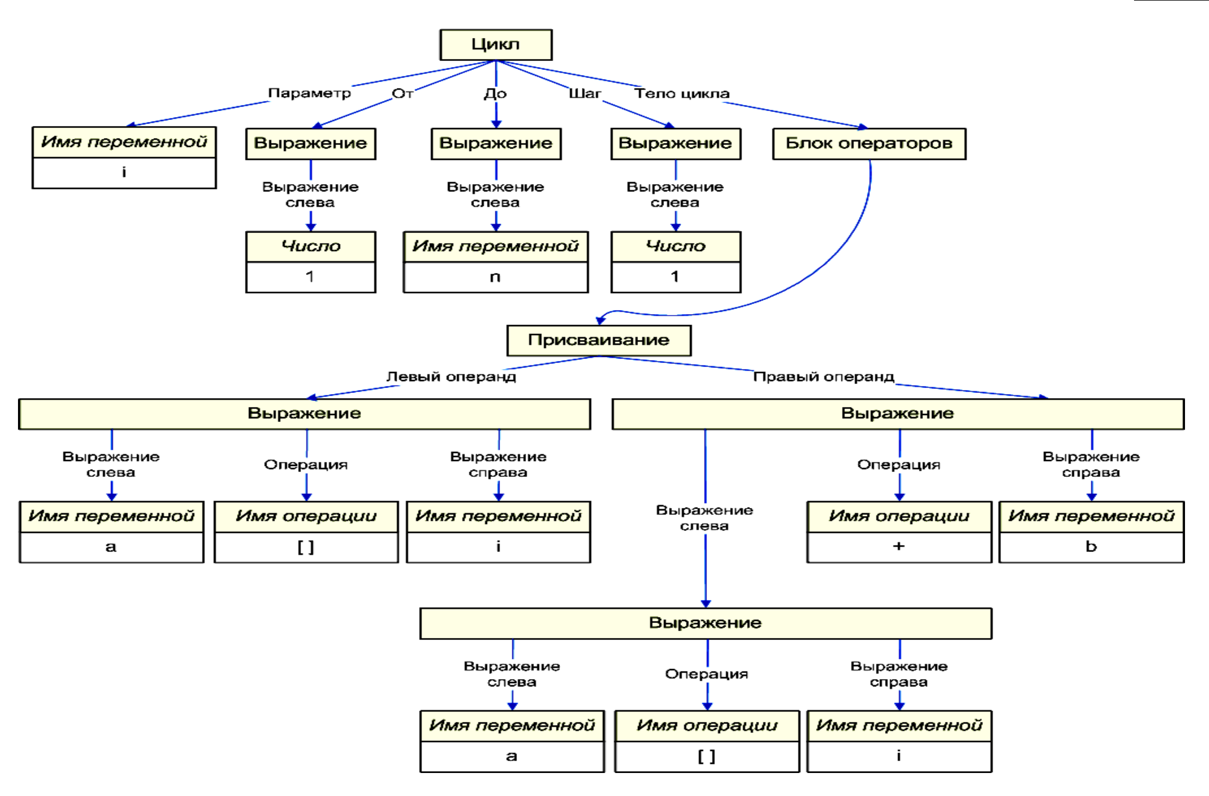


Рисунок 2 – Вход метода интерпретации – модель структурной программы

В процессе выполнения метода всем фрагментам МСП типа «выражение» добавляются три атрибута: аргументное, результатное и сильнорезультатное множества (рис. 3).

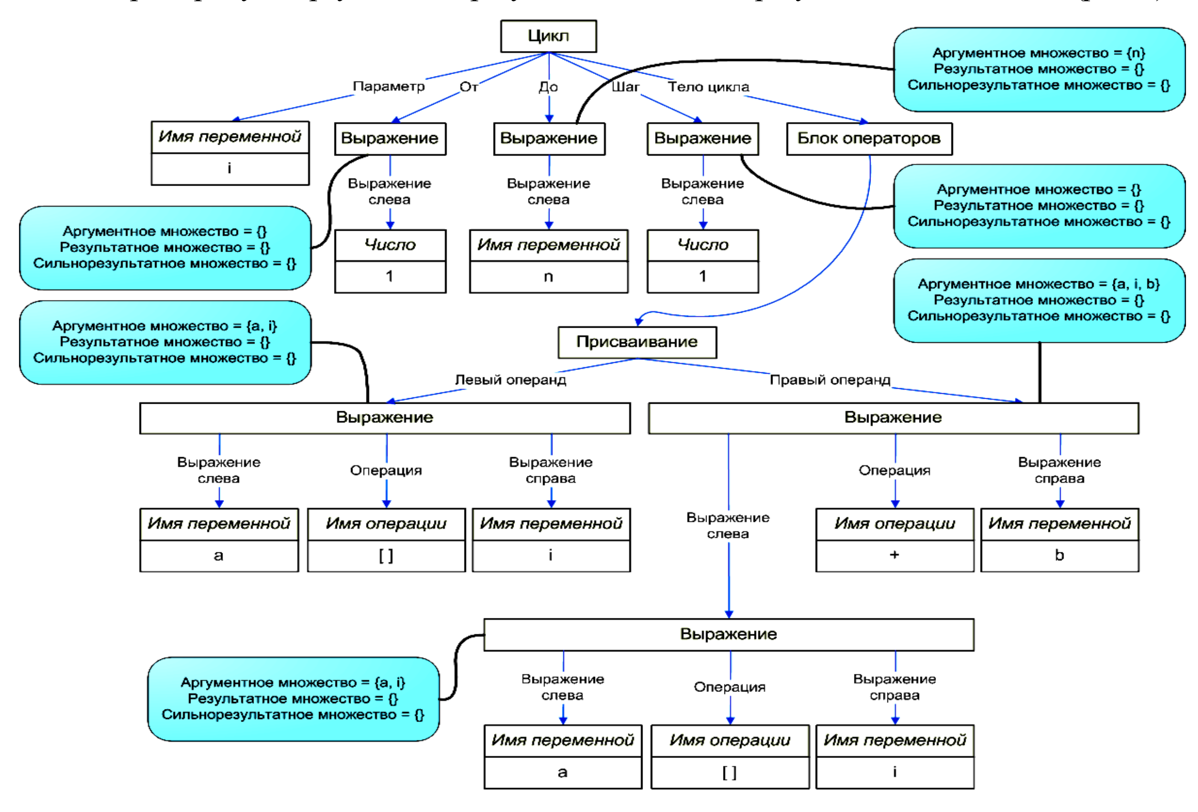


Рисунок 3 – Выход интерпретатора – модель структурной программы, расширенная терминами потокового анализа

Подсистема интерпретации методов потокового анализа программ в Специализированном банке знаний о преобразованиях программ

Подсистема потокового анализа программ является частью программного наполнения СБкЗ_ПП, архитектура которого представлена на рис. 4 [9], [10]. СБкЗ_ПП является распределенной системой, работающей по технологии клиент-сервер. На серверной части находится информационное наполнение банка знаний, содержащее знания о преобразованиях программ, и оболочка информационного наполнения, предоставляющая доступ к этим знаниям. На клиентской части работает преобразователь программ и редакторы знаний о преобразованиях программ. Рассмотрим архитектуру преобразователя программ.

Входной информацией подсистемы потокового анализа, управляемого знаниями, являются: модель структурной программы, методы потокового анализа программ, модель онтологии потокового анализа и задание на потоковый анализ программ. Входная информация формируется подсистемой управления СБкЗ_ПП, которая осуществляет взаимодействие с редакторами и передает управление подсистеме потокового анализа. На выходе подсистемы формируется МСП, расширенная терминами потокового анализа программ.

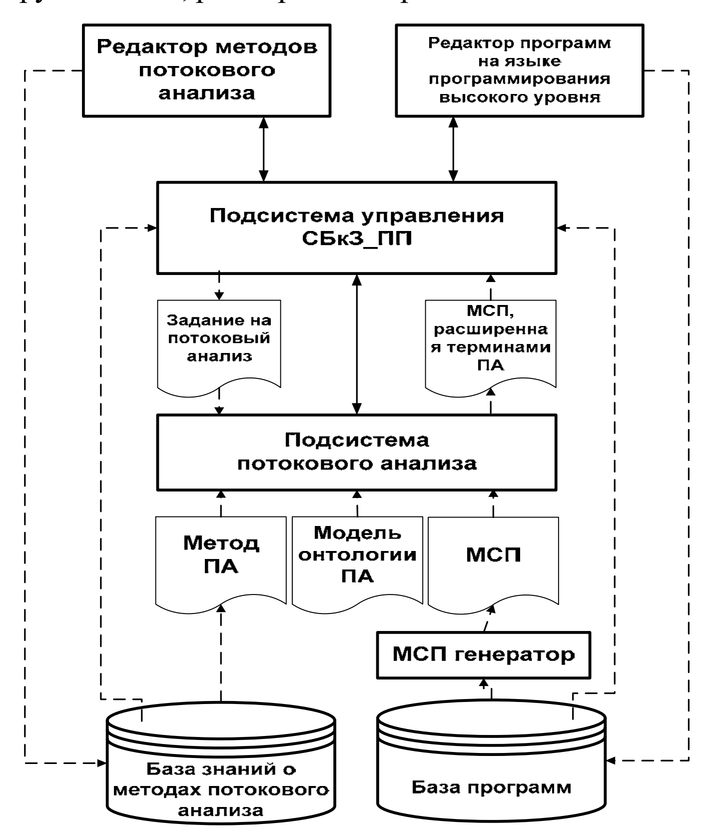


Рисунок 4 – Архитектура СБкЗ_ПП

Предлагаемый в работе подход отличается тем, что термины оптимизации программ закреплены и описаны в виде модели онтологии.

В подсистеме потокового анализа выделено пять блоков (рис. 5):

1) *выбор метода ПА* – получает задание на потоковый анализ и определяет, какие методы потокового анализа программ и в каком порядке должны быть применены к МСП;

2) *проверка метода ПА на соответствие онтологии* – проверяет метод ПА на валидность. Проверка производится на соответствие онтологии;

3) *исполнение инструкций метода потокового анализа* – обходит дерево метода потокового анализа программ снизу-вверх и слева-направо и вызывает соответствующий ей обработчик понятия онтологии потокового анализа. Вызов обработчика производится через специальный компонент – *Менеджер безопасности*. После завершения обхода дерева метода потокового анализа, модель структурной программы подается на выход;

4) *менеджер безопасности* – загружает необходимые обработчики понятий онтологии потокового анализа из *репозитория обработчиков понятий онтологии потокового анализа*. Ограждает систему от несанкционированных действий обработчика. Обработчик «живет» в рамках жестко заданного контекста. Обработчику позволяется:

– читать и изменять модель структурных программ, поданную на вход подсистеме потокового анализа;
– считывать версию Java™ платформы;
– читать и изменять глобальные переменные;

5) *репозиторий обработчиков понятий онтологии потокового анализа* – хранилище программных модулей, каждому из которых сопоставляется некоторое подмножество понятий онтологии потокового анализа программ.

Множество методов, с которым может работать интерпретатор, не является фиксированным. Все методы хранятся в базе знаний о методах потокового анализа программ и описаны в терминах онтологии знаний о методах потокового анализа программ. Редактор методов потокового анализа программ позволяет изменять содержимое базы знаний. Процесс редактирования знаний в редакторе осуществляется в терминах онтологии знаний о преобразованиях программ [11].

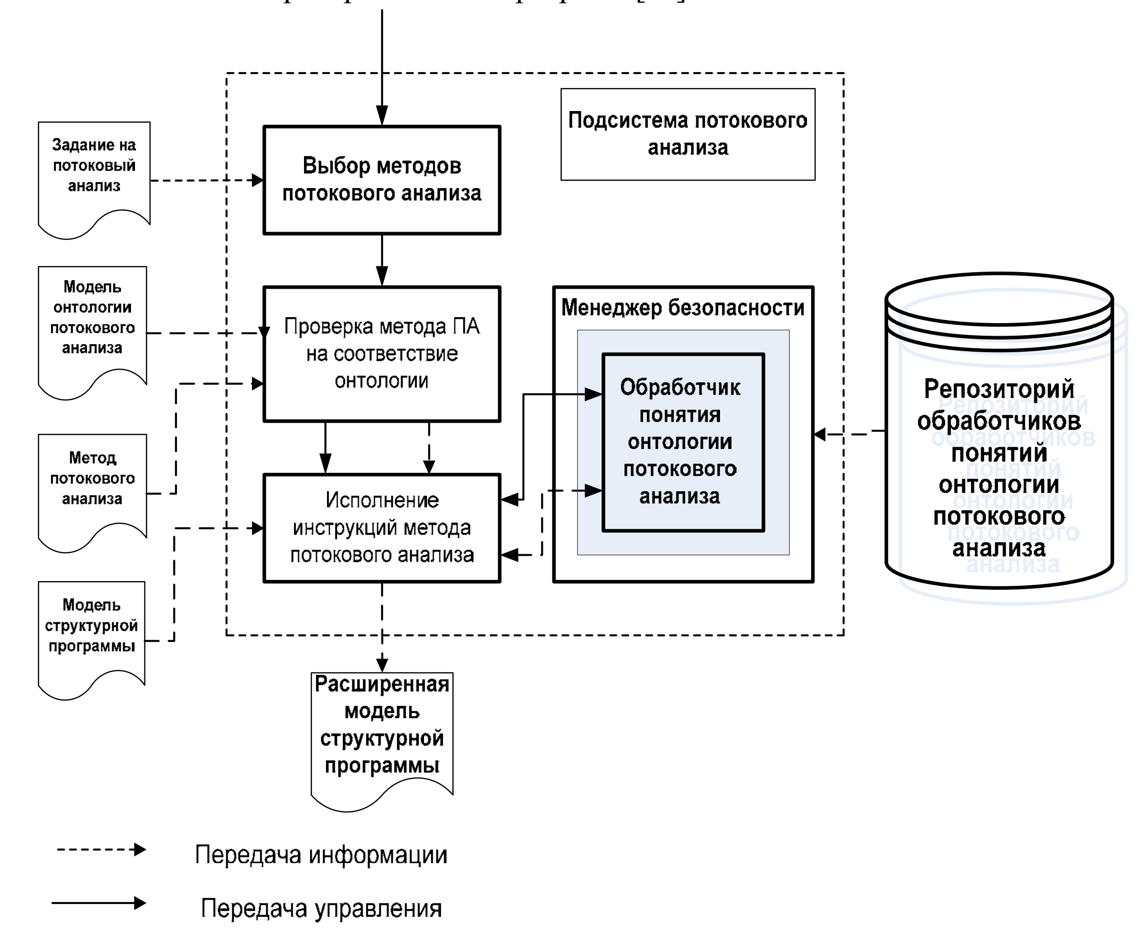


Рисунок 5 – Архитектура подсистемы потокового анализа программ

Заключение

В работе выполнен анализ существующих оптимизирующих и распараллеливающих компиляторов и предложен подход к представлению методов потокового анализа программ и к выполнению интерпретации методов потокового анализа программ, представленных в виде МСП. Методы реализованы в специализированном банке знаний о преобразованиях программ. Представлена концепция потокового анализа, управляемого знаниями. На примере было продемонстрировано, каким образом при помощи языка методов потокового анализа можно представлять методы потокового анализа. На основе концепции потокового анализа, управляемого знаниями, разработана подсистема потокового анализа в рамках системы преобразований программ в СБкЗ_ПП. Система реализована под платформу Java™ с использованием ресурсов многоцелевого банка знаний (http://www.iacp.dvo.ru/es/mpkbank).

Литература

1. Математические модели и методы в параллельных процессах. – М.: Наука, Гл. ред. физ.- мат. лит., 1986.
2. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб.: БХВ – Петербург, 2002. – 608 с.
3. Евстигнеев В.А. Касьянов В.Н. Инструментальная система для изучения преобразований программ // Интеллектуализация и качество программного обеспечения. – Новосибирск: ИСИ СО РАН, 1994. – С. 90-99.
4. Штейнберг Б.Я. Математические методы распараллеливания рекуррентных циклов для суперкомпьютеров с параллельной памятью. – Ростов н/Д: Изд-во Рост. ун-та, 2004. – 192 с.
5. Касьянов В.Н., Трахтерброт М.Б. Анализ структур программ в глобальной оптимизации // Труды Всесоюзного симпозиума по методам реализации новых алгоритмических языков. – Ч.1. – С. 143-160.
6. Князева М.А., Купневич О.А. Модель онтологии предметной области «Оптимизация последовательных программ». Определение языка модели структурных программ // НТИ. – Сер. 2. – 2005. – № 2. – С. 17-21.
7. Князева М.А., Волков Д.А. Потоковый анализ программ, управляемый знаниями // Программные продукты и системы. – 2007. – № 1. – С. 49-52.
8. Князева М.А., Купневич О.А. Модель онтологии предметной области «Оптимизация последовательных программ». Определение расширения языка модели структурных программ терминами потокового анализа // НТИ. – Сер. 2. – 2005. – № 4. – С. 14-22.
9. Клещев А.С., Князева М.А. Управление информацией о преобразованиях программ. I. Анализ проблем и пути их решения на основе методов искусственного интеллекта // Изв. РАН. ТиСУ. – 2005. – № 4. – С. 118-127.
10. Орлов В.А., Клещев А.С. Компьютерные банки знаний. Универсальный подход к решению проблемы редактирования информации // Информационные технологии. – 2006. – № 5. – С. 25-31.
11. Москалев И.И. Разработка и реализация редактора знаний о потоковом анализе программ // Тез. Докл. XXXIII Дальневосточной математической школы-семинара им. акад. Е.В. Золотова 29 августа-4 сентября 2008. – Владивосток: Дальнаука. – 2008.

M.A. Knyazeva, I.I. Moskalev

Interpretation methods of flow analysis of program in Specialized knowledge bank of program transformations.

The concept of Specialized knowledge bank of program transformations is developed in order to solve educational, scientific and practical problems in the field of program transformations according to a new scientific direction “Computer knowledge banks”. The modern point of view on existing program transformation systems is given in this paper. Then we suggest the approach to development of program transformation system within the Specialized knowledge bank of program transformations.

Статья поступила в редакцию 10.07.2008.